



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/712,655

11/13/2003

Altan J. Stalker

A-8121

8962

62658

7590

09/17/2009

MERCHANT & GOULD
SCIENTIFIC ATLANTA, A CISCO COMPANY
P.O. BOX 2903
MINNEAPOLIS, MN 55402-0903

EXAMINER

ALSIP, MICHAEL

ART UNIT

PAPER NUMBER

2186

MAIL DATE

DELIVERY MODE

09/17/2009

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/712,655	Applicant(s) STALKER, ALTAN J.	
	Examiner MICHAEL ALSIP	Art Unit 2186	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 10 July 2009.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-12 and 18-28 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-12 and 18-28 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>8/28/2009</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 1-6, 9, 10, 12, 18-23, 26, and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Judge et al. (US 6,430,570 B1), as applied to **claims 1 and 18** above, and further in view of Enterprise JavaBeans Component Architecture: Designing and Coding Enterprise Applications, Henceforth referred to as EJB.

3. Consider **claims 1 and 18**, Judge et al. discloses a system and method for managing memory, the system and method comprising a memory with logic, and receiving a application state from each of a plurality of applications in memory, wherein each said application includes a plurality of operational stages (abstract, Col. 4 lines 55-

Art Unit: 2186

67, Col. 5 lines 1-15 and Col. 7 lines 20-27); and determining which of the plurality of applications to effect removal from the memory based on a received application state (Col. 4 lines 55-67, Col. 5 lines 1-15, Col. 7 lines 28-51, and Col. 9 lines 3-11).

Judge et al. discloses the ability to set or change the order in which applications are unloaded in case of a low or no memory condition (Col. 7 lines 28-51 and Col. 8 lines 22-30 and lines 54-58), but does not explicitly state wherein each application state indicates the presence of differences between an operational stage of a corresponding activated application at a point when memory is evaluated for application removal and the operational stage of the corresponding application upon being unloaded from the memory and reactivated, nor does Judge et al. explicitly state wherein an application with an application state indicating an absence of said differences between said activated and reactivated operational stages is removed from the memory before other applications with application states indicating the presence of said differences between said activated and reactivated operational stages, however EJB teaches the use of stateless and stateful applications and the stateless applications have only two states, 'ready' and 'does not exist', with this type of application there is no difference when the application is terminated then reactivated. The removal of stateless applications before other types of applications, and by doing so causing an application with no differences when removed and reactivated to be removed from memory before other types of applications, is beneficial because there is no overhead with respect to removing and reloading the application and no data needs to be moved from memory to secondary memory, therefore reducing latency in the system (pg. 4 ¶'s 2 and 7, therefore removing

Art Unit: 2186

an application with a stateless state before other types of applications reduces latency in the system and provides better performance for the user), EJB also discloses the newly added claim limitation: **wherein said receiving an application state from each of the plurality of applications in memory includes receiving a stateful state with a state record indicating the absence of said differences between said activated and reactivated operational stages and no significant ones of said user perceivable differences between said activated and reactivated application**

(Judge: Col. 7 lines 52-65, where the application manager saves a state of the application before unloading it from memory therefore having a stateful state with a state record, the fact that the application manager stores the state of the application, indicates that the application manager is notified that a state needs to be saved therefore indicating a stateful state with state record, also the stateless state is disclosed as well. When the stateful state has a state record it is indicating that, because of this state record being saved, that there will now be no perceivable differences between said activated and reactivated applications.).

It would have been obvious to one of ordinary skill in the art at the time of the invention to remove a stateless state application before other types of applications in the system of Judge et al., because EJB discloses that there is no overhead with respect to removing and reloading the application and no data needs to be moved from memory to secondary memory, therefore reducing latency in the system (pg. 4 ¶s 2 and 7).

Art Unit: 2186

4. Consider **claim 12**, Judge et al. discloses a method for managing memory, said method comprising the steps of: receiving an indication that memory space is needed in memory, wherein each said application includes a plurality of operational stages; (abstract, Col. 4 lines 55-67, Col. 5 lines 1-15, Col. 7 lines 28-51, and Col. 9 lines 3-11), wherein said receiving an application state includes receiving a stateless state indicating the absence of said differences between said activated and reactivated operational stages and no significant ones of user perceivable differences between said activated and reactivated application, a stateful state with a state record state indicating the absence of said differences between said activated and reactivated operational stages and no significant ones of said user perceivable differences between said activated and reactivated application, and a stateful state with no state record indicating the presence of differences between said activated and reactivated operational stages and the presence of said user perceivable differences between said activated and reactivated application (Col. 7 lines 52-65, where the application manager saves a state of the application before unloading it from memory therefore having a stateful state with a state record, the fact that the application manager stores the state of the application, indicates that the application manager is notified that a state needs to be saved therefore indicating a stateful state with state record.).

Judge et al. discloses the ability to set or change the order in which applications are unloaded in case of a low or no memory condition (Col. 7 lines 28-51 and Col. 8 lines 22-30 and lines 54-58), but does not explicitly state wherein each application state indicates the presence of differences between an operational stage of a corresponding

Art Unit: 2186

activated application at a point when memory is evaluated for application removal and the operational stage of the corresponding application upon being unloaded from the memory and reactivated, however EJB teaches the use of stateless and stateful applications and an application being stateless or stateful indicates the presence of differences between an application that is unloaded and then reactivated (pg. 4 ¶'s 2 and 7).

Judge et al. also does not explicitly state determining which of the plurality of applications to effect removal from the memory based on the received single indication for each of the plurality of applications in memory, wherein the step of determining includes the steps of determining that an application with a stateless state is removed before an application with a stateful state with a state record, and that a stateful state with a state record is removed before a stateful state with no state record; and effecting the removal of an application with a stateless state before the removal of an application with a stateful state with a state record, and effecting the removal of an application with a stateful state with a state record before the removal of an application with a stateful state with no state record, however EJB teaches that stateless applications have better performance due to the fact that no data is stored back and forth to secondary memory therefore freeing up resources that a stateful application would require if it were stored to and from secondary memory (pg. 4 ¶'s 2 and 7, therefore removing an application with a stateless state before an application with a stateful state would reduce latency in the system and provide better performance for the user. The examiner is considering, for the purpose of this claim, that the stateful state with and state record and the stateful

Art Unit: 2186

state without a state record to be synonymous, because the state record is not recorded until the data is to be unloaded, therefore before the unload procedure the stateful state has no state record and afterwards the stateful state has a state record, both being the same stateful state application, therefore the stateful state always has a record before being removed. In other words, when the application state has not yet been saved it is indicative of a difference between the active and reactivated application, but when the state is finally saved the differences are then absent).

It would have been obvious to one of ordinary skill in the art at the time of the invention to remove the application with the stateless state before an application with a stateful state in the system of Judge et al., because EJB discloses that this reduces latency and overhead in the system (pg. 4 ¶'s 2 and 7).

5. Consider **claims 2 and 19**, as applied to **claims 1 and 18** above, Judge et al. in view of EJB discloses wherein said receiving an application state from each of a plurality of applications in memory includes receiving one of a stateless state indicating the absence of said differences between said activated and reactivated operational stages and no significant ones of user perceivable differences between said activated and reactivated application and a stateful state with no state record indicating the presence of differences between said activated and reactivated operational stages and the presence of said user perceivable differences between said activated and reactivated application (Col. 7 lines 52-65 and EJB, pg.'s 3-4 section 2.4 Enterprise JavaBeans, the stateless state is disclosed. Where the claim language only requires an indication of one of the above states and the stateless state is disclosed).

Art Unit: 2186

6. Consider **claims 3 and 20**, as applied to **claims 2 and 19** above, Judge et al. discloses all the limitations of **claims 2 and 19** above, and also discusses the use of a Java application manager, but does not explicitly state the applications having a stateless state. EJB teaches stateless state applications are an integral part of Java systems and that stateless state applications lend to better performance in the system by freeing up resources and being scalable for a large number of clients (pg. 1 section: Enterprise JavaBeans (EJB), pg.'s 3-4 section 2.4 Enterprise JavaBeans) therefore being obvious to one of ordinary skill in the art at the time of the invention.

7. Consider **claims 4 and 21**, as applied to **claims 1 and 18** above, Judge et al. in view of EJB discloses wherein said receiving a stateful state with a state record includes receiving a state that indicates a user would perceive no significant difference between a presentation associated with one of the plurality of applications before and after removal from the memory and reloading to the memory because the state is saved in the state record (Judge et al.: Col. 7 lines 52-65 EJB: Stateful bean).

8. Consider **claims 5 and 22**, as applied to **claims 4 and 21** above, Judge et al. discloses further including effecting the removal of the application with a stateful state with a state record and saving the state record (Col. 7 lines 52-65).

9. Consider **claims 6 and 23**, as applied to **claims 5 and 22** above, Judge et al. discloses further including, responsive to a user activating the removed application, restoring the removed application with the saved state record (Col. 7 lines 52-65).

10. Consider **claims 9 and 26**, as applied to **claims 2 and 19** above, Judge et al. discloses the ability to set or change the order in which applications are unloaded in

Art Unit: 2186

case of a low or no memory condition (Col. 7 lines 28-51 and Col. 8 lines 22-30 and lines 54-58), but does not explicitly state wherein said determining which of the plurality of applications to effect removal includes determining that an application with a stateless state is removed before an application with a stateful state with a state record, and that a stateful state with a state record is removed before a stateful state with no state record, however EJB teaches that stateless applications have better performance due to the fact that no data is stored back and forth to secondary memory therefore freeing up resources that a stateful application would require if it were stored to and from secondary memory (pg. 4 ¶'s 2 and 7, therefore removing an application with a stateless state before an application with a stateful state would reduce latency in the system and provide better performance for the user. The examiner is considering, for the purpose of this claim, that the stateful state with and state record and the stateful state without a state record to be synonymous, because the state record is not recorded until the data is to be unloaded therefore before the unload procedure the stateful state has no state record and afterwards the stateful state has a state record, both being the same stateful state application, therefore the stateful state always has a record before being removed).

It would have been obvious to one of ordinary skill in the art at the time of the invention to remove the application with the stateless state before an application with a stateful state in the system of Judge et al., because EJB discloses that this reduces latency and overhead in the system (pg. 4 ¶'s 2 and 7).

Art Unit: 2186

11. Consider **claims 10 and 27**, as applied to **claims 2 and 19** above, Judge et al. discloses the ability to set or change the order in which applications are unloaded in case of a low or no memory condition (Col. 7 lines 28-51 and Col. 8 lines 22-30 and lines 54-58), but does not explicitly state further including effecting the removal of an application with a stateless state before the removal of an application with a stateful state with a state record, and effecting the removal of an application with a stateful state with a state record before the removal of an application with a stateful state with no state record, however EJB teaches that stateless applications have better performance due to the fact that no data is stored back and forth to secondary memory therefore freeing up resources that a stateful application would require if it were stored to and from secondary memory (pg. 4 ¶'s 2 and 7, therefore removing an application with a stateless state before an application with a stateful state would reduce latency in the system and provide better performance for the user. The examiner is considering, for the purpose of this claim, that the stateful state with and state record and the stateful state without a state record to be synonymous, because the state record is not recorded until the data is to be unloaded therefore before the unload procedure the stateful state has no state record and afterwards the stateful state has a state record, both being the same stateful state application, therefore the stateful state always has a record before being removed).

It would have been obvious to one of ordinary skill in the art at the time of the invention to remove the application with the stateless state before an application with a

Art Unit: 2186

stateful state in the system of Judge et al., because EJB discloses that this reduces latency and overhead in the system (pg. 4 ¶s 2 and 7).

12. **Claims 7, 8, 11, 24, 25 and 28** are rejected under 35 U.S.C. 103(a) as being unpatentable over Judge et al. (US 6,430,570 B1) in view of Enterprise JavaBeans Component Architecture: Designing and Coding Enterprise Applications, Henceforth referred to as EJB.

13. Consider **claims 7 and 24**, as applied to **claims 2 and 19** above, Judge et al. in view of EJB discloses all the limitations of **claims 2 and 19** above, but does not explicitly state wherein said receiving an indication of a stateful state with no state record includes receiving a state that indicates a user would perceive a difference between a presentation associated with one of the plurality of applications before and after removal from the memory and reloading to the memory, however the examiner is taking official notice to the fact that receiving a stateful state with no state record is common and well-known. As an example, when a word document is being closed by a user before having been saved (stateful state with no state record), the user is prompted with information asking the user if they wish to save their unsaved data (yes or no) or cancel the closing of the application, where the user is given the option to select no therefore receiving an indication of a stateful state with no state record, therefore indicating to the user that there will be a perceived difference.

It would have been obvious to one of ordinary skill in the art at the time of the invention for the user to receive an indication about information pertaining to the state of an application the user is using when that application is to be removed, because

Art Unit: 2186

providing the user with options and information with respect to the applications the user is using allows for better user control, flexibility and user experience and to better manage the unintentional loss of user data.

14. Consider **claims 8 and 25**, as applied to **claims 7 and 24** above, Judge et al. does not explicitly state wherein **said** receiving a stateful state with no state record includes receiving unload information, wherein the unload information includes at least one of an unload information explanation and unload information choices, however the examiner is taking official notice to the fact that receiving an indication of a stateful state with no state record is common and well-known. As an example, when a word document is being closed by a user before having been saved (stateful state with no state record), the user is prompted with unload information choices pertaining to whether the user wishes to save their unsaved data (yes or no) or cancel the closing of the application.

It would have been obvious to one of ordinary skill in the art at the time of the invention for the user to be prompted with information pertaining to the state of an application the user is using and provide options to the user when that application is to be removed, because providing the user with options and information with respect to the applications the user is using allows for better user control, flexibility and user experience and to better manage the unintentional loss of user data.

15. Consider **claims 11 and 28**, as applied to **claims 1 and 18** above, Judge et al. discloses all the limitations of **claims 1 and 18** above, but does not explicitly state further including providing an explanation to a user when an application to be removed

Art Unit: 2186

from the memory includes a stateful state with no state record, wherein the explanation informs the user the result of removing the application, the examiner is taking official notice to the fact that when an application is removed from memory, it is common for the user to be prompted with information informing the user the result of removing the application. As an example, when a word document is being closed by a user before having been saved (stateful state with no state record), the user is prompted with information asking the user if they wish to save their unsaved data (yes or no) or cancel the closing of the application, thereby providing the user with information informing the user the result of removing the application.

It would have been obvious to one of ordinary skill in the art at the time of the invention for the user to be prompted with information pertaining to the state of an application the user is using when that application is to be removed, because providing the user with options and information with respect to the applications the user is using allows for better user control, flexibility and user experience and to better manage the unintentional loss of user data.

Response to Arguments

16. Applicant's arguments filed 7/10/2009 have been fully considered but they are not persuasive.

17. The Applicant argues that Judge or EJB disclose the limitation added to claim 1, which is a previously present claim limitation: **wherein said receiving an application state from each of the plurality of applications in memory includes receiving a stateful state with a state record indicating the absence of said differences**

between said activated and reactivated operational stages and no significant ones of said user perceivable differences between said activated and reactivated application. However, as rejected in the claim language itself, Judge discloses in Col. 7 lines 52-65, that the application manager saves a state of the application before unloading it from memory therefore having a stateful state with a state record, the fact that the application manager stores the state of the application, indicates that the application manager is notified that a state needs to be saved therefore indicating a stateful state with state record and EJB discloses in pg.'s 3-4 section 2.4 Enterprise JavaBeans the use of stateful beans, therefore this previously presented limitation is disclosed by the above two references. When the stateful state has a state record it is indicating that, because of this state record being saved, that there is now no perceivable differences between said activated and reactivated applications. The Applicant also argues that EJB does not disclose that the states of the sessions are saved when the client terminates a session, however not only does Judge already disclose this feature, but EJB, as the Applicant admits, discloses doing so when the bean is saved and restore to and from memory and secondary memory, the moving of data to a secondary memory is the termination of the session, applications are not executed from secondary memory, they are executed from the main memory.

18. The Applicant also argues that the Examiner as conceded that the combination of Judge and EJB fails to disclose "a stateful state with no state record indicating the presence of said differences between said activated and reactivated operation stages and the presence of said user perceivable differences between said activated and

Art Unit: 2186

reactivated applications". The Examiner has not conceded that the combination does not disclose this limitation, but rather, with respect to this claim that the Examiner is considering, for the purpose of this claim, that the stateful state with and state record and the stateful state without a state record to be synonymous, because the state record is not recorded until the data is to be unloaded, therefore before the unload procedure the stateful state has no state record and afterwards the stateful state has a state record, both being the same stateful state application, therefore the stateful state always has a record before being removed. In other words, when the application state has not yet been saved it is indicative of a difference between the active and reactivated application, but when the state is finally saved the differences are then absent.

Conclusion

19. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Art Unit: 2186

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MICHAEL ALSIP whose telephone number is (571)270-1182. The examiner can normally be reached on Monday through Friday 7:30AM to 5:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matt Kim can be reached on 571-272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Michael Alsip
Examiner
Art Unit 2186

/Michael Alsip/
Examiner, Art Unit 2186

/Matt Kim/
Supervisory Patent Examiner, Art Unit 2186

September 15, 2009